Enabling Flow-level Reliability on FTDMA Schedules with efficient Hop-by-hop Over-provisioning

Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, Fabrice Valois

Enabling Flow-level Reliability on FTDMA Schedules with efficient Hop-by-hop Over-provisioning

Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, Fabrice Valois

ISSN 0249-6399 ISRN INRIA/RR--8866--FR+ENG

RESEARCH REPORT

N° 8866

February 2016

Project-Team UrbaNet

Enabling Flow-level Reliability on FTDMA Schedules with efficient Hop-by-hop Over-provisioning

Guillaume Gaillard *, Dominique Barthel†, Fabrice Theoleyre‡, Fabrice Valois §

Project-Team UrbaNet

Research Report n° 8866 — version 2 — initial version February 2016 — revised version April 2016 — 19 pages

Abstract: The industrial Internet of Things (IoT) relies on efficient and reliable communications from the source devices to the network gateways. The challenge resides in providing reliable multi-hop radio paths. Indeed, even in the case of synchronized nodes following a Frequency-Time Division Multiple Access (FTDMA) schedule, the radio links suffer from interference and packet losses. Resource allocation algorithms on FTDMA must take into account the requirements of the applications in terms of delivery.

We propose two allocation mechanisms enabling retransmissions, first uniform, and second hop by hop. They give each flow on the network the possibility to satisfy its applicative end-to-end delivery constraint. By reducing the amount of resource necessary for retransmissions, and balancing the allocations on the relay nodes, we provide a robust and efficient resource allocation.

In order to validate our approach, we implement our mechanisms enabling retransmissions on top of the state of the art algorithm, Traffic-Aware Scheduling Algorithm (TASA). By the means of simulations, we show the gains of our proposals in terms of reliability, and their cost in terms of number of allocations.

Key-words: IoT, multi-hop, scheduling, FTDMA, delivery ratio, reliability, retransmissions, TASA

* Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France. guillaume.gaillard1@insa-lyon.fr

† Orange Labs R&D, Meylan, France. dominque.barthel@orange.com

‡ ICube, Université de Strasbourg / CNRS, France. theoleyre@unistra.fr

§ Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France. fabrice.valois@insa-lyon.fr

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Fiabilisation des échéanciers FTDMA à l'aide de retransmissions saut par saut efficaces

Résumé : Dans un environnement industriel, la collecte d'information pour l'Internet des Objets (IoT), par le biais de réseau radio multi-saut est sujette au problème de la fiabilité. En effet, même dans le cas où les noeuds sont synchrones et suivent un échéancier de type Frequency-Time Division Multiple Access (FTDMA), les liens subissent de l'interférence et des pertes de paquets. Les algorithmes d'allocation de ressources sur FTDMA doivent prendre en compte les contraintes de fiabilité attendues par les différentes applications.

Nous proposons deux mécanismes d'allocation permettant des retransmissions, d'abord uniforme, puis saut par saut. Ils permettent à chaque flux présent sur le réseau de valider sa contrainte de taux de livraison applicatif et bout-en-bout. En réduisant le nombre de ressources nécessaires aux retransmissions, et en les répartissant sur les noeuds relais, nous rendons notre allocation fiable et efficace en énergie.

Afin de valider notre approche, nous intégrons nos mécanismes de prise en compte des retransmissions dans l'algorithme qui fait référence dans l'état de l'art, Traffic-Aware Scheduling Algorithm (TASA). Nous montrons au moyen de simulations les gains de nos approches en termes de fiabilité, et leurs couts en termes de quantité de ressources allouées.

Mots-clés : IoT, multi-saut, allocation de ressources, FTDMA, taux de livraison, fiabilité, retransmissions, TASA

1 Introduction

1.1 Context: on the interest of FTDMA scheduling

The data gathering in the Internet of Things requires reliable and efficient radio communications. The operators need to demonstrate that they can satisfy a robust Quality of Service (QoS) and adapt to several demands. Indeed, each application has specific requirements. The traffic is divided into client flows, and each flow is separately serviced. This way, the operators are able to provide multi-fow guarantees.

The architecture must provide guarantees concerning the reliability. They must correctly deliver the applicative messages of the different flows to the gateways. The solution must be energy efficient. At the IETF, the 6TiSCH working group [8] provides such a technology where packets are relayed hop-by-hop on synchronized nodes (the relays).

The transmissions are scheduled in time and frequency blocks or cells (thus forming a FT-DMA, each cell being a combination of a time slot and a channel offset). They avoid collisions: no pair of cells interferes in a 6TiSCH schedule. The example of a FTDMA schedule presented in Fig. 1 is based on a minimal topology where all the relay nodes are neighbors: the cells can not be spatially re-used. In the general case, a set of 2D schedules would be necessary to represent all the cells.

The cells are the minimal resource we allocate, they correspond to the transmission of a frame and the reception of the corresponding acknowledgment. The frames are typically less than 127 Bytes large and the time slot is 10ms long.

6TiSCH gives the possibility to associate each cell with a *track*. Each track corresponds to a flow with specific QoS constraint. Frames of a given flow shall only use the transmission cells that are labeled with the corresponding track ID (denoted inside the cells in Fig. 1a). Hence, some resource is dedicated to each flow. This feature enables flow isolation.

The Traffic-Aware Scheduling Algorithm (TASA) represents a pioneering piece of work to centrally compute a FTDMA schedule [6]. TASA gives optimal schedule in terms of schedule compactness (see Fig. 1a). TASA focuses on traffic generated locally on the distributed nodes. TASA is a relevant centralized solution for scheduling in 6TiSCH networks.

1.2 Problem statement, proposals

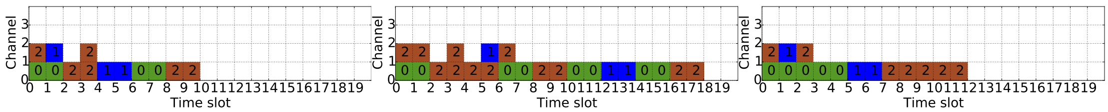
Still, in a real word, packets suffer losses. Radio transmissions between sensor nodes are subject to collisions, fading, external interference, that negatively impact the Packet Error Rate (PER) of each radio link [2].The end-to-end Packet Delivery Ratio (PDR) of a flow depends on the PER of each hop in the path to a gateway: the QoS is consequently degraded.

TASA does not take into account the reliability concerns. The packet delivery ratio is not the priority. TASA provides optimal schedules in terms of resource consumption [6], given a routing topology, but TASA does not take into consideration:

- the robustness to packet losses;
- the adaptability to an increase of traffic.

Here we propose to allocate additional cells for frame retransmissions. The radio retransmissions increase the probability of success of a transmission. They enable to satisfy a given Packet Delivery Ratio (PDR), but they impact the energy consumption by increasing the quantity of allocated resource.

Dobslaw et al propose SchedEx [3], a schedule extension that complements a schedule in order to guarantee a minimal network level end-to-end reliability. The authors calculate the number



(a) TASA schedule. (b) TASA schedule with uniform over-provisioning. (c) TASA schedule with hop-by-hop over-provisioning.

Figure 1: Example of FTDMA Schedule shapes built by TASA and by the proposed extensions. Based on a tree topology with 3 leaves, 3 relays, 1 gateway. Each leaf sends 1 message of a given application. The blue app. message is a single frame while the two other app. have messages of 2 fragments. The track IDs are shown in the cells.

of necessary retransmissions for all the packets, at each link of the routing tree. This expected number of retransmissions is defined according to the load of a radio link and its reliability. In other words, Schedex does not guarantee flow isolation with differentiated PDR requirements.

With our solution, we give a different number of hop-by-hop cells, added to the schedule for the retransmissions, for each hop and flow. Hence, we can satisfy multiple reliability constraints (one for each flow). This fits with the notion of tracks in 6TiSCH: every frame in a node's queue will transit to the next hop by choosing the cells according to its track ID. This way the flow isolation is respected and the frames are retransmitted using the calculated allocations.

In this work, we address the problem of the reliability in terms of PDR by allocating additional resource for the retransmission of frames. This over-provisioning complements the assignment. We first consider a uniform over-provisioning mechanism: the number of cells added for the retransmissions is constant for all the links in a path to a gateway. This enables to reach a given PDR but with a significant overhead. Secondly, we provide a hop-by-hop mechanism that maintains the PDR along each path, and adapts the number of cells added for the retransmissions to the PER of each link and to the allocation load on the nodes. We integrate both mechanisms in TASA and compare the results trough simulations.

Fig. 1b and Fig. 1c illustrate the extension of the TASA schedule in 1a with added cells for retransmissions, according to the two approaches. For the same number of frames to deliver, a different number of cells is allocated for the retransmissions in order to satisfy the PDR.

Our contribution is four-fold:

1. we provide a first mechanism to calculate the minimum number of cells, uniformly allocated on each link of a path, that enable enough retransmissions to satisfy the expected delivery ratios;

2. we provide a second mechanism that calculate the minimum number of cells, distributed hop-by-hop, that enable enough retransmissions to satisfy the expected delivery ratios;

3. we implement the two allocation mechanisms on a TASA schedule to enhance its performance.

4. we compare the performance of TASA and our proposals in terms of PDR satisfaction, and cost in terms of number of allocations.

2 Model

We model here the network characteristics that we consider for the scheduling algorithms. In order to construct the FTDMA schedules, the resource allocation algorithm manipulates variables (e.g. transmissions, frames, queues) and objects that mimic the behavior of the considered network. Table 1 summarizes the parameters and notations we use in the model. We further detail their description.

2.1 Topology

We model a sensor network as a set of nodes exchanging messages in a multi-hop way. The nodes have a single half-duplex radio interface and have buffering capacities. 3 distinct types of nodes are considered:

- the leaf nodes which only generate the traffic;
- the intermediary routers that forward it;
- one or several gateways that collect it.

Each leaf node carries one or several applications that generate messages of constant size (one or several frames). For each application, a leaf node creates one flow of messages directed to a gateway.

We use the Packet Error Rate (PER) to characterize link quality. The PER represents the ratio of unsuccessful transmissions of frames on a link. Both classical propagation models and empirical results give an estimation of the relationship between the PER and distance, attenuation, and type of node [4]. The reality often differs from the model: a PER can depend on other parameters such as interference, noise, state of the nodes, etc [2]. We assume that a monitoring mechanism provides frequent updates on the link qualities that make possible the schedule adaptation. We adopt the simplified path loss model presented in [4].

We consider that a node B is a neighbor of another node A if the PER of link AB is not 100%. We consider that the PER is time invariant on the scale of the scheduling, and that it does not depend on the channel offset.

2.2 Communications and resource allocation

Communications follow a FTDMA schedule. The FTDMA schedule is a matrix of time slots (of constant duration) and frequency channels (e.g. 16). One allocation is an assignment of a time-frequency block, named cell, to the transmission of one fragment between two neighboring nodes.

Depending on its size, each message is divided into one or several fragments. Each fragment separately transits on a single cell. Hence each applicative message is considered as a set of fragments, their quantity depending on the size of the message.

Typically, a time slot of 10ms can carry a 127 B frame. A larger message would be fragmented, into e.g. 3 frames, requiring one cell per fragment. The model does not imply a forwarding strategy (route-over or mesh-under).

We isolate each flow: every allocated cell is assigned to one flow only. The set of cells assigned to one flow is named track. Each track is identified with a track ID.

We consider a centralized scheduling algorithm that takes as input parameters two pieces of topological information:

1. a routing acyclic graph. For each flow, we build a loop-free path from the leaf node to a gateway. In the case of TASA, the routing graph may be computed by using RPL information [9]. RPL is a distributed tree routing algorithm that lets each router select the parent that minimizes the cumulated ETX metric to the root. Our model is also applicable to multi-path routing.

Table 1: Parameters for the model of over-provisioning

Variable	Explication
$f \in F$	A flow f among the set of flows F
$path(f)$	The path of f
$< l_1, l_2, \dots, l_{gw} >$	The set of links of the path of f
$n_{msg}(f)$	Number of messages in a slotframe from f
$n_{frag}(f)$	Number of fragments in an applicative message
$PDR^{min}_{msg}(f)$	Constraint: minimum end-to-end PDR for the messages from f
$PDR_{msg}(f)$	End-to-end PDR for the messages from f
$PDR^{hop}_{msg}(f, l)$	PDR of the messages from f at link l.
$pdr^{min}_{frag}(f)$	Constraint: expected end-to-end PDR for the fragments from f
$pdr_{frag}(f)$	End-to-end PDR of the fragments from f, without retransmissions
$pdr^{rtx}_{frag}(f, k)$	End-to-end PDR of the fragments from f, with k transmissions
$per(l)$	Packet Error Rate (PER) on a given link l
$n_{cell}(l)$	Number of allocated cells on a given link l
$alloc_f$	Set of allocation counts along the path, for one message of f
$alloc^{l_i}_f$	i_{th} component of $alloc_f$: Allocation count of the i_{th} link of $path(f)$
$load(l)$	Allocation load on a given link l
	Number of over-provisioning cells (OPC) for one message from f:
$n^{unif}_{opc}(f)$	added by the uniform over-provisioning mechanism
$n_{opc}(f, l)$	added by the hop-by-hop over-provisioning mechanism at link l
$n^{max}_{rtx}(f)$	Maximum number of retransmissions per hop, for one message from f
$Sol(f)$	Set of solutions of the hop-by-hop algorithm

2. the Packet Error Rate (PER) of every link. A set of links interfere if there is a loss of information when transmissions take place at the same time slot and channel. We need to build the conflict graph, i.e. the information about the set of interfering links.

   In case no other information is given about interference, we build the conflict graph by considering that the three-hop neighbors and beyond are not interferers.

2.3 Expression of the reliability constraint

Each application has its own reliability constraint, expressed as the expected applicative end-to-end Packet Delivery Ratio (PDR), denoted $PDR^{min}_{msg}(f)$. For each flow f, the PDR is the ratio between the number of messages received at the gateway, and the number of messages sent from the source. A schedule satisfies the reliability constraint if the minimum PDR is respected over a long period (e.g one day).

We consider that an applicative message is lost when at least one of its fragments is lost. Under the hypothesis that the fragment transmissions are not correlated, we easily deduce from the applicative PDR the required PDR associated to one fragment in Eq. (1).

$$\forall f \in F,\ \left(pdr^{min}_{frag}(f)\right)^{n_{frag}(f)} = PDR^{min}_{msg}(f) \tag{1}$$

In Eq. (1), F is the set of flows, $n_{frag}(f)$ the number of fragments of the messages of flow f, $pdr^{min}_{frag}(f)$ the expected fragment PDR and $PDR^{min}_{msg}(f)$ the applicative PDR of flow f (cf. Table 1).

2.4 Traffic definition

We consider an heterogeneous traffic, mixing several applications [1]. We make the assumption that the traffic of each application is bounded by a periodical amount of fragments. We address the allocation of resource for periodical traffic patterns. The FTDMA schedule is hence divided into periodical slotframes (set of time slots) that repeat in time, indefinitely.

From the perspective of the allocation algorithm, all the nodes have buffered the packets at the beginning of the slotframe. The fragment must have reached a gateway during the same slotframe.

Each node is capable of detecting the loss of a fragment (if it does not receive an acknowledgment). In this case, it retransmits the fragment to the next-hop neighbor on the next available cell associated with the same track ID. Both the technical implementation and the schedule algorithm limit the number of hop-by-hop retransmissions for a message to a maximum per hop (e.g. 15 fragments). Note that in our model, this maximum applies to each message (set of fragments) and not to each fragment.

In a first approach we consider uniform over-provisioning along the path of a flow. At each hop, a constant number of cells is added in order to let the nodes retransmit a fragment when a first transmission failed. From the perspective of the allocation algorithm, we choose to model the retransmissions as duplicates of a fragment of a message, generated at the source leaf node of each flow. They transit over all the path like the fragments. With this assumption, the scheduler covers the case where the fragments are retransmitted a constant number of times at each node.

In a second approach we consider hop-by-hop allocations. In this case, from the perspective of the allocation algorithm, we choose to model the retransmissions as *local* duplicates of a fragment of a message. In our model, they are generated, at each hop, on the transmitting node, and they disappear on the reception node. With this assumption, the scheduler covers the case where the fragments are retransmitted a variable number of times at each node.

The cells added by the resource allocation algorithm to make possible the retransmissions are named *over-provisioning cells (OPC)* (Table 1).

3 A first approach: a uniform over-provisioning

In this section, we provide a method to calculate the minimal number of cells $n_{opc}^{unif}(f)$ that, uniformly added to the schedule along a path, enables to assert the PDR requirements.

3.1 Expression of the fragment end-to-end PDR with retransmissions

We can easily calculate $pdr_{frag}(f)$, the end-to-end PDR of a fragment without retransmissions for a flow f in Eq. (2).

$$pdr_{frag}(f) = \prod_{l \in path(f)} (1 - per(l)) \tag{2}$$

In Eq. (2), $path(f)$ is the set of links that bind the source to one gateway. Without retransmission opportunities, the schedule does not ensure that $pdr_{frag}(f) \geq pdr_{frag}^{min}(f)$ and hence that the PDR is met for flow f.

Eq. (3) gives $pdr_{frag}^{rtx}(f,k)$ the end-to-end PDR of a fragment with k transmission opportunities ($k-1$ over-provisioning cells) for a flow f.

$$pdr_{frag}^{rtx}(f,k) = 1 - (1 - pdr_{frag}(f))^k \tag{3}$$

Eq. (3) expresses the success of at least one delivery, with k independent intents.

3.2 Satisfying the reliability constraint with retransmissions

In order to calculate the minimum number of over-provisioning cells needed for a message, we associate the OPC with the fragments of the message. We uniformly assign the over-provisioning cells to the fragments of the message. We divide $n_{opc}^{unif}(f)$ by $n_{frag}(f)$ as following:

$$n_{opc}^{unif}(f) = q \cdot n_{frag}(f) + r, \quad 0 \leq r < n_{frag}(f) \text{ with } n_{opc}^{unif}(f) \leq n_{rtx}^{max}(f) \tag{4}$$

In Eq. (4), q and r are the quotient and remainder in the Euclidean division of $n_{opc}^{unif}(f)$ by $n_{frag}(f)$. This way, each fragment obtains q or $q+1$ associated retransmission opportunities (Eq. (3)). In other words, each fragment may be transmitted $q+1$ or $q+2$ times.

The transmission of a message corresponds to the transmissions of all its fragments (the success of the former is the product of the successes of the others). From the expression of the delivery constraint for fragments (Eq. (1)) and the previous section (Eq. (2) and Eq. (3)), we compute the end-to-end PDR for the messages of each flow f, $PDR_{msg}(f)$ in Eq. (5).

$$PDR_{msg}(f) = \left(pdr_{frag}^{rtx}(f, q+1)\right)^{n_{frag}(f)-r} \left(pdr_{frag}^{rtx}(f, q+2)\right)^{r} \tag{5}$$

And developping with Eq. (3):

$$PDR_{msg}(f) = \left(1 - (1 - pdr_{frag}(f))^{q+1}\right)^{n_{frag}(f)-r} \left(1 - (1 - pdr_{frag}(f))^{q+2}\right)^{r} \tag{6}$$

In Eq. (6), $pdr_{frag}(f)$ is the end-to-end PDR of a fragment without retransmissions (Eq. (2)). In Eq. (5), $pdr_{frag}^{rtx}(f, k)$ is the end-to-end PDR of a fragment with k transmissions (Eq. (3)).

Our objective is to find the minimal $n_{opc}^{unif}(f)$ enabling to repect the reliability constraint:

$$PDR_{msg}(f) \geq PDR_{msg}^{min}(f) \tag{7}$$

Starting with 0 over-provision cell, $n_{opc}^{unif}(f) = 0$, we iteratively increment $n_{opc}^{unif}(f)$ until we obtain the minimal one that gives Eq. (7) or we reach the maximum number of retransmissions $n_{rtx}^{max}(f)$. In this case, the PDR constraint is not satisfied. We set $n_{opc}^{unif}(f)$ at maximum value and proceed with the next flow.

4 A second approach: computing hop-by-hop over-provisioning

We now provide a method to compute the hop-by-hop minimal number of retransmission opportunities for each link l of a given path, that enables to meet the PDR requirements. Our approach reduces the total number of cell allocations over the path, while satisfying the applicative end-to-end PDR.

4.1 Expression of the per-message end-to-end PDR with retransmissions

Our constraint remains the same as in Eq. 7, but in the following we express $PDR_{msg}(f)$ with hop-by-hop over-provisioning cells.

As previously, we consider each hop as independent and each transmission of a fragment also as independent. We now have several numbers of over-provisioning cells along the path, so the Eq. (2), giving the end-to-end delivery of a fragment does not suit anymore.

For each given hop, the successful transmission of a message depends on the successful transmissions of its fragments. At least one success should occur for each fragment of the message, all of them within the available opportunities. A successfully transmitted fragment is not retransmitted. In other words, we require at least $n_{frag}(f)$ successes among $alloc_f^l$ intents (cf. Table 1 for the notations). This condition is expressed by the cumulative distribution function of a Bernoulli formula in Eq. (8):

$$\forall f \in F,\ \forall l \in path(f)$$
$$PDR_{msg}^{hop}(f,l) = \sum_{k=0}^{n_{opc}(f,l)} \binom{alloc_f^l}{k} per(l)^k (1 - per(l))^{alloc_f^l - k}$$
$$\text{with}\ \ alloc_f^l = n_{opc}(f,l) + n_{frag}(f) \quad (8)$$

In Eq. (8), $PDR_{msg}^{hop}(f,l)$ is the hop-by-hop PDR for one message of a given flow f at link l.

Considering each hop independent from the others, our constraint can be expressed as the product along the path of each hop expression:

$$\forall f \in F,\quad PDR_{msg}(f) = \prod_{l \in path(f)} PDR_{msg}^{hop}(f,l) \quad (9)$$

4.2 Satisfying the reliability constraint with minimum hop-by-hop retransmissions

Our goal is to find the best balanced set of over-provisioning cells enabling to respect the reliability constraint (Eq. (7)).

If $alloc_f$ is the set of allocation counts along the path, for a flow f, then:

$$alloc_f \;=\; \{alloc_f^l\}_{l \in path(f)} \quad (10)$$

$alloc_f$ is considered valid if all allocation numbers $alloc_f^l$ remain between the number of fragments and the maximum number of transmissions:

$$\forall l \in path(f),\ n_{frag}(f) \le alloc_f^l \le n_{frag}(f) + n_{rtx}^{max}(f) \quad (11)$$

We denote as $Sol(f)$ the set of $alloc_f$ that verifies Eq. (7) and Eq. (11). $Sol(f)$ is the set of solutions of the approach.

Given the assumption that the number of retransmissions for the fragments of each message is limited to a maximum (Eq. (11)), our goal is to distribute the allocations on the links network-wide. The allocation load is the sum of the already allocated cells for a given link, with the cells allocated for f:

$$\forall l \in path(f),\ load(l) = \left(n_{cell}(l) + n_{msg}(f) \cdot alloc_f^l\right) \quad (12)$$

In order to reduce the overall differences between the numbers of cell allocations, we look for minimizing the maximum link load over each path:

$$\min_{alloc_f \in Sol(f)} \left(\max_{l \in path(f)} load(l) \right) \quad (13)$$

We propose an inverse greedy algorithm that gives a solution to this problem. The algorithm is applied flow by flow. The number of *iterations* of our proposal is equal to the length of the path. At each iteration, one link is chosen and treated: its allocation count is saved.

At first iteration $i = 0$**:** For every message and at each link of the path, whatever the flow is, we bound the number of fragment retransmissions to the maximum $n_{rtx}^{max}(f)$. At the

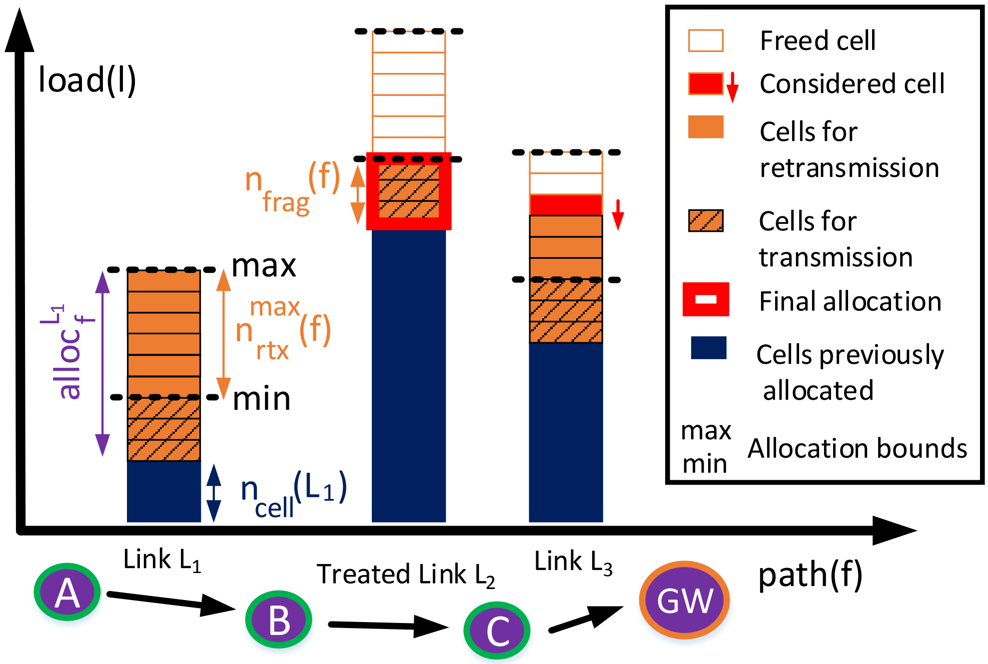


Figure 2: Composition of the load of the links of a path.

beginning of the algorithm, for the considered flow f, the number of allocated cells is initialized at each hop using the maximum value, with $n_{rtx}^{max}(f)$ retransmissions:

$$\forall l \in path(f),\ alloc_f^l \leftarrow n_{frag}(f) + n_{rtx}^{max}(f) \tag{14}$$

In this state, if the PDR computed by using Eq. (9), leads to an unsatisfied constraint in Eq. (7), $Sol(f)$ is empty, there is no solution. The algorithm proceeds with another flow.

In the other cases, if the PDR meets the requirements (Eq. (7)), then **for each iteration** i**:** the algorithm follows two steps:

1. We determine the most loaded link $l_{max}(f)$ of the path:

$$\text{Find } \{l_{max}(f) \in path(f)\} \quad \text{such as} \quad \left\{ load(l_{max}(f)) = \max_{l \in path(f)} load(l) \right\} \tag{15}$$

   In Eq. (15), $load(l)$ is the load of the link l of the considered path.

   Fig. 2 illustrates Eq. (15) and Eq. (14), for one path over 3 links. The values of the components of $alloc_f$ are bounded between the number of fragments of a message (3 in the figure) and the maximum number of transmissions ($6 + 3 = 9$ in the figure).

2. The allocation count $alloc_f^{l_{max}(f)}$ is decremented for the link $l_{max}(f)$. Then, the retransmission count $n_{opc}(f, l_{max}(f))$ is also decremented accordingly.

Stop conditions of iteration i**:** We repeat these two steps until the algorithm stops for one link l:

- the allocation count $alloc_f^l$ reaches the number of fragments $n_{frag}(f)$;

- **or:** the condition Eq. (7) is no longer respected.

At the end of the iteration, the minimal allocation value has been found for link l. We save the value, mark the link as treated, it and *iterate* the algorithm with the other links of the path.

Final iteration: When the numbers of retransmissions have been computed for all the links over the path, we update the number of allocated cells for each node with the new values. The algorithm proceeds with the next flow.

Proof of minimality: **At the end of the first iteration of the algorithm**, one link denoted as l_0 is treated: it reaches a minimum number of allocations (Eq. (10)), that enables to satisfy Eq. (7). Indeed, one of the stop conditions has been reached.

Because the algorithm always decrements the link which has the maximum load value (Eq. (12)), l_0 is the first treated link and its load is higher than the load of the other links. There are two possibilities:

- l_0 was always the most loaded link: in this case, the allocation counts of the other links are at their maximum (as in initial condition, Eq. (14)). Hence no additional retransmission can be added to any other link to compensate the impact of a decrement of the load of l_0. Hence, we found the minimal maximum load value for l_0.

- l_0 became at a given time the most loaded link: necessarily, the other untreated links that had previously got the maximum load, have at most a difference of one decrement with l_0 (otherwise they would have been treated). If we had not decremented them, then maybe we could decrement l_0 once or twice more, and still satisfy Eq. (7), but in any case one another link would have a maximum load value. Hence we also found the minimal maximum load value for l_0.

At each following iteration, the remaining links have lower load value. Otherwise, they would have been treated earlier (step 1 of the algorithm). Hence, the final solution satisfies Eq. (7) and the objective Eq. (13).

□

Note: The variable $PDR_{msg}(f)$ (Eq. (9)) decreases with each decrement in the number of cells (step 2 of the algorithm). But the impact of a decrement depends on the PER of each link. An algorithm aiming at reducing the overall number of cells would choose to decrement first the load of the links with the highest impact on Eq. (7) (i.e. the links by increasing order of PER).

5 Adding over-provisioning for the retransmissions: Application to the TASA algorithm

The Traffic-Aware Scheduling Algorithm [5] [7] represents a key reference for centralized FTDMA scheduling. In this section, we briefly remind the behavior of TASA, then we show how we can implement our two approaches and consequently improve TASA's performance.

5.1 Scientific background: the Traffic-Aware Scheduling Algorithm (TASA)

TASA focuses on the traffic generated at each node in the network. The algorithm considers that every node has a FIFO structure named *queue*. A given amount of frames (denoted as *packets*) is present in the queues of the leaf nodes at the beginning of the schedule (slot 0). At every allocation, a packet moves from the transmitter queue to the receiver queue.

TASA allocates as many cells in a schedule as the expected traffic during a time period, and repeats the resulting slotframe indefinitely in time. The approach gives optimal schedule in terms of schedule length.

TASA takes as input parameters:

- one routing tree rooted at each gateway;
- a conflict graph.

The schedule is build slot after slot. TASA iteratively applies a *matching technique* at each slot:

1. at slot k, the algorithm recursively selects the links that will get a cell allocated. The selection process crosses the trees from the roots to the leaves, and chooses the links according to the load and respecting the half-duplex constraint on any node (Listing 1). The set of selected links can be used at the same time slot (no node is both transmitter and receiver at slot k). The set is call Duplex Conflict-Free Link set ($DCFL(k)$).
2. Then, TASA applies a coloring function at each slot k. The algorithm assigns a channel offset c for each subset of links of $DCFL(k)$ that do not interfere one another (the subsets are determined easily with the conflict graph).
3. The queues are updated: the cell (k, c) is allocated to the oldest frames in the queues.

Because TASA builds schedules considering each fragment independent one another, a route-over forwarding mechanism does not suit. Indeed, with route-over, a node has to wait for all the fragments of a message before forwarding it. Since TASA does not consider the precedence for the transmission cells, the order cannot be guaranteed.

In Listing 1, n is the number of recursions. The sub-tree load in step 2 is the sum of the queue lengths in the tree rooted at the considered node. The step of recursion (step 4) enables to satisfy the half-duplex constraint.

```
{n=0}
Set of roots: the set of gateways.

1) Search for the set of nearest descendants with a non empty queue;
2) Selection of the descendant Ni with the maximum sub tree load.
3) The link between Ni and his parent Pi is included in the DCFL(k);
4) The selection recursively proceeds at {n+1} with the following set of roots:
 – the other children of Pi, if any;
 – the children of Ni, if any.

Ending condition: the leaf nodes have no descendant.
```

Listing 1: Recursive Selection of Duplex Conflict-Free Link (DCFL) sets in TASA.

5.2 Implementing the uniform over-provisioning approach in TASA

In the basic TASA setup, no retransmissions are considered. We propose to implement the uniform approach by considering end-to-end duplicates of the fragments. We run the TASA scheduler without any modification.

The change resides in the traffic definition of the algorithm. At slot 0, each leaf node, source of a flow, now has in its queue:

1. the fragments of each applicative message;
2. the end-to-end duplicates associated with the fragments of each message (Eq. (5)).

This way, TASA allocates cells for a constant number of retransmissions all along the path. The resulting schedule provides enough cells at each hop for the satisfaction of the PDR. In a network running the 6TiSCH stack, the nodes detect the loss of a frame when they do not receive acknowledgment. In this case, they intent a retransmission on the next cell having the same track ID. The same behavior is repeated until they reach the maximum number of retransmissions. The fragment is then dropped and removed from the buffer.

5.3 Implementing hop-by hop over-provisioning in TASA

In order to integrate the hop-by-hop over-provisioning cells of the second approach into TASA, we slightly modify the original algorithm:

1. When summing the traffic of a sub-tree, we take into account the necessary number of over-provisioning cells $n_{opc}(f,l)$ that we calculated (Eq. (10)) along with the length of the queues;
2. The selection function includes the over-provisioning cells (the changes appear in Listing 2);
3. The cell allocation order changes. The cell (k,c) is allocated:

   to the oldest fragment in the queue;

   if previous allocation was for the last fragment of a message, to the local duplicates, $n_{opc}(f,l)$ times;
4. The queues are updated by message after the allocation of the last over-provisioning cell. All the fragments are transferred to the receiver's queue.

These modifications enable a route-over mechanism, because they provide consecutive cells that give every node the possibility to collect all the fragments before forwarding them again.

The differences in the select function are described in Listing 2.

{n=0}
Set of roots: the set of gateways.

1) Search for the set of nearest descendants with a non empty queue **(including over−provisioning cells)**;
2) Selection of the descendant Ni with the maximum sub tree load, **(including over−provisioning cells)**.
3) The link between Ni and his parent Pi is included in the DCFL(k);
4) The selection recursively proceeds at {n+1} with the following set of roots:
 − the other children of Pi, if any;
 − the children of Ni, if any.

Ending condition: the leaf nodes have no descendant.

Listing 2: Recursive Selection of (DCFL) sets in TASA with hop-by-hop retransmissions.

6 Evaluation

In this section, we compare by simulation:

- TASA(original);
- TASA with uniform over-provisioning cells ($TASA_{unif}$);
- TASA with hop-by-hop over-provisioning cells ($TASA_{hbh}$).

6.1 Scenario

Based on an ad-hoc network simulator in Python, we run simulations on a given topology. The simulations emphasizes the impact of the variations of given parameters, the others kept at default. Table 2 summarizes the parameters and values.

The nodes are placed in a rectangle of 400*200 meters. The relays are placed on a triangle mesh (every approximately 70 meters). The leaf nodes are uniformly spread on the rectangle. The two gateways are placed at positions (100,100) and (300,100).

We define two different applications with given traffic pattern and constraint. Each leaf node support one flow, half of them pertaining to the first application, the other part to the second application.

We apply a path loss propagation model [4] and compute the PER for each link. Fig. 3 shows the distribution of PERs according to the distance between the node and the type of node.

For the need of simplicity, two nodes are considered neighbors if both their PERs are lower than a given threshold : 95%.

Based on the PER values, we construct a routing tree rooted at each gateway, according to the ETX metric. We build the conflict graph based on the 2-hops neighborhood.

We compare TASA and our two extensions over 10 randomly generated topologies that abide by these characteristics, according to 10 values of the considered parameter. We consider variations of the following parameters:

1. The **traffic intensity**: we show the behavior of the mechanism at the limit of the capacity;
2. The **slotframe size**: similarly, when the schedules are too long to fit in the slotframe, only a portion of them applies, so the constraints are not satisfied;
3. The **expected PDR constraint** itself: we highlight the limit of the algorithm in terms of PDR satisfaction;
4. The **maximum number of authorized retransmissions** at each hop for a message: this parameter limits the hop-by-hop allocation mechanism but corresponds to a concrete case (the MAC layer protocol, or the implementation, is restricted to this constraint in order to limit the delays and the risk of buffer overflow);

We evaluate the performance of our approach according to three criteria:

1. The satisfaction of the PDR constraint for each flow: we evaluate the reliability of the created schedules with the percentage of successful flows;
2. The maximum number of allocations on a node: this criteria details the capacity of the network to support more traffic;
3. The number of allocations in the network: we evaluate the efficiency of the solutions with the global allocation of resources.

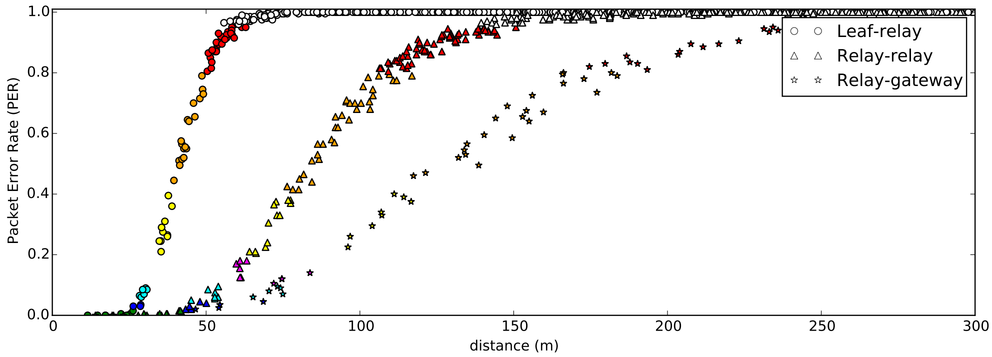
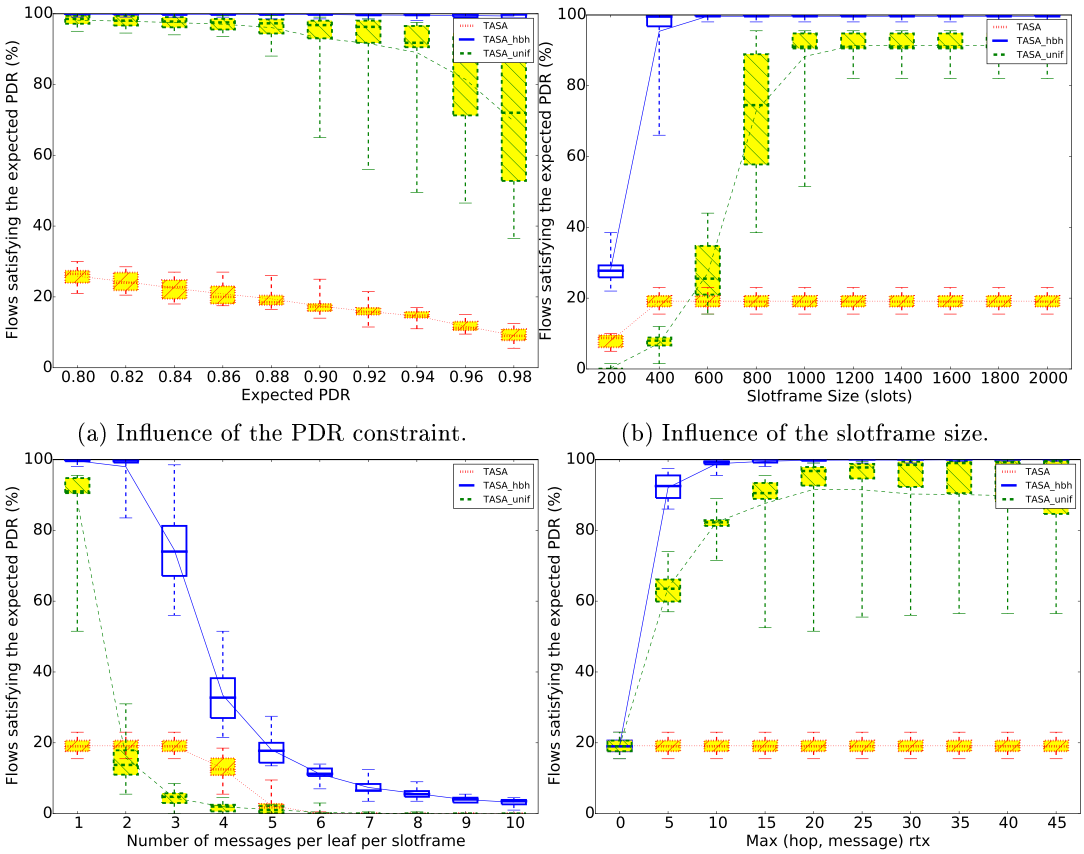


Figure 3: Distribution of Packet Error Rates (PER) by role type.

Table 2: Parameters of the simulation scenario.

Parameter	default value	range in simulation
Size of slotframe	1000 slots	200 to 2000 slots
Number of channels	16	
Maximum number of retransmissions	16	0 to 45
Number of leaf nodes	200	
Number of relay nodes	24	
Number of message per slotframe, app. 1	1	1 to 10
Number of message per slotframe, app. 2	1	1 to 10
Number of fragments per message, app. 1	3	
Number of fragments per message, app. 2	2	
Expected end-to-end PDR, app. 1	0.97	0.80 to 0.98
Expected end-to-end PDR, app. 2	0.80	0.80 to 0.98



(a) Influence of the PDR constraint.

(b) Influence of the slotframe size.

(c) Influence of the traffic load.

(d) Inf. of the max. number of retransmissions.

Figure 4: Evaluation of the performance in terms of PDR satisfaction.

6.2 Results

The following figures highlight the differences between the original schedule and the extensions with retransmissions. Each boxplot represents the minimum, first quartile, median, third quartile and maximum values obtained for the 10 topologies. The line binding the boxplots represents the mean values.

Fig. 4 shows that the original scheduler (TASA) does not permit to satisfy the reliability constraint on lossy links. The two over-provisioning mechanisms enable the satisfaction of PDR constraints, in cases where the network limit of capacity is not yet reached. In the latter case, the hop-by-hop algorithm behaves better: its performance degrades less rapidly. TASA does not take into account the PDR constraint. In our simulation, around 40 flows (20%) luckily satisfy their PDR constraint without any retransmissions (Fig. 4d). In Fig. 4a, this proportion decreases from 25% to 10%.

In the latter figure we highlight a rupture in the performance of the uniform mechanism. Because of the significant number of over-provisioning cells, the length of the schedule exceeds the size of the slotframe (1000). The cells that do not fit in the slotframe are not considered, hence, the PDR is degraded.

Similarly, the performance is clearly degraded for short slotframe sizes (Fig. 4b). The uniform mechanism keeps under the hop-by-hop one because its assumptions are more demanding: by considering the independence of the transmissions of the fragments, it over-estimates the number of needed cells.

TASA serves more flows than $TASA_{unif}$ in two situations:

1. with a slotframe of 400 slots in Fig. 4b;
2. with 3 or 4 messages per leaf per slotframe, in Fig. 4c.

In both situations, with $TASA_{unif}$ the slotframe is fully field with over-provisioning cells, at the expense of classical cells that TASA dedicates to other flows.

Fig. 5 shows that the uniform over-provisioning mechanism is expensive in terms of network capacity. In Fig. 5a, the number of cells allocated to the most loaded relay node, for transmission or reception, is constant for TASA: TASA does not consider the delivery constraint. With the hop-by-hop mechanism, the schedule remains acceptable (the maximum load is always less than twice the value with the original algorithm). The number of over-provisioning cells slowly increases for low values of the PDR constraint. Then, for the values of 0.94 and beyond, the satisfaction of the constraint requires more cells and the cost in allocations rapidly increases.

The uniform mechanism rapidly saturates the most loaded relay node because of the excessive number of over-provisioning cells. A mean value of 700 allocated cells is obtained for large slotframe sizes (from 1000 to 2000 slots in Fig. 5b, limited by the default max. number of retransmissions) or for large maximum numbers of retransmissions (from 25 to 45 in Fig. 5d limited by the default slotframe size).

The network is rapidly saturated (the most loaded relay is occupied at 100%) under the influence of the traffic load (Fig. 5c). With high traffic intensity (e.g. 3 messages per leaf) the schedule length reaches the slotframe size (1000). From this point the increase in the number of allocations for the most loaded node is less important because the schedule is cut to fit in a slotframe.

Finally, Fig. 6 shows a significant impact of the uniform mechanism on the number of allocated cells in the network. This number has similar evolution to the maximum per node. We note that with the default parameters, there is around 4000 allocated cells for the uniform mechanism, (Fig. 6b), 2000 for the hop-by-hop mechanism, (Fig. 6d) and 1000 for TASA. This differences are due to the use on all the network of over-provisioning cells. At each slot, the DCFL set respectively has an average size of 1, 2, and 4 links for the three algorithms.

In Fig. 6b, the length of the schedule built by $TASA_{hbh}$ is less than 600 slots. From this value, the influence of the slotframe size on the performance is null.

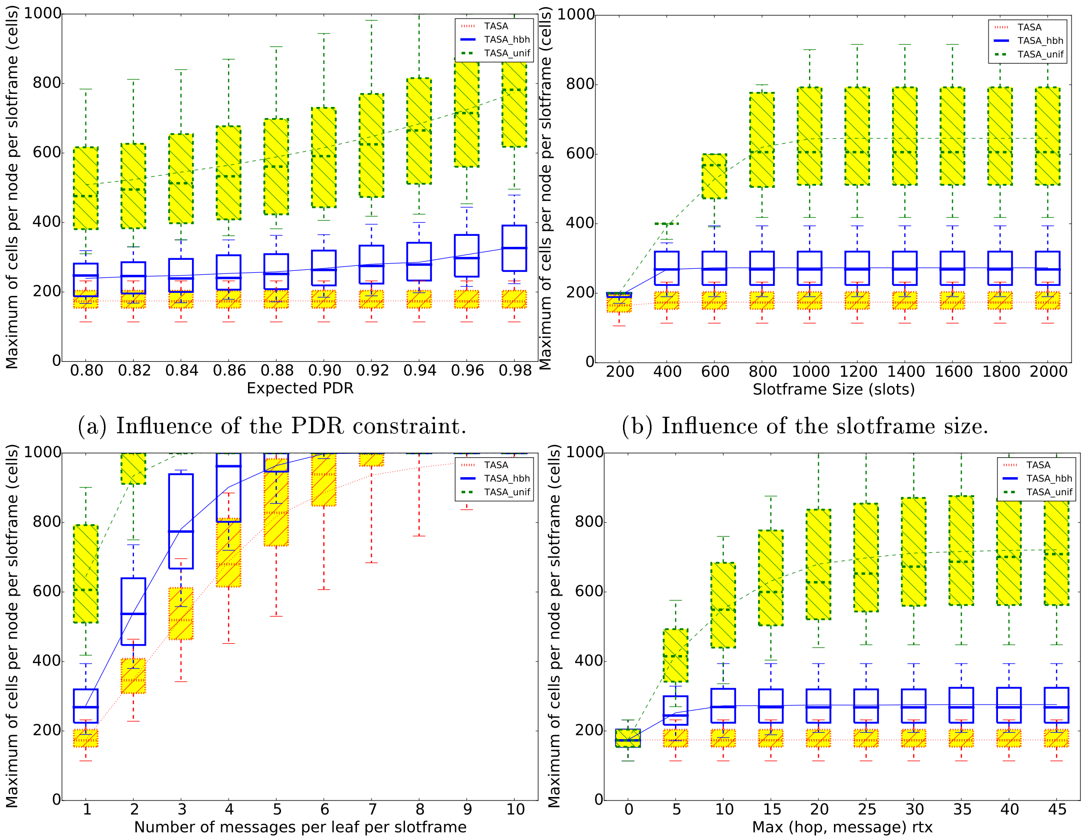
In Fig. 6c, the changes in the direction of the lines are explained by the fact that at this point (3 messages per leaf per slotframe), we reach the limit of capacity of the network. The schedule is larger than the slotframe and some allocations are not considered. The number of allocated cells still increases with the traffic intensity, because more cells are allocated at the beginning of the schedule (mainly from the first hops).

7 Conclusion

In a network where multiple applications have different delivery constraints, the operator must offer differentiated QoS in terms of reliability.

In this work, we provide an efficient way to give reliability to FTDMA schedules. We propose two mechanisms of over-provisioning that adapt to an expected end-to-end packet delivery ratio taking as parameter the quality of each link in a path.

We implement the two mechanisms in TASA. Our results show that we enhance the scheduling algorithm by providing reliability on lossy links while limiting the overhead in terms of allocations.



(a) Influence of the PDR constraint.

(b) Influence of the slotframe size.

(c) Influence of the traffic load.

(d) Inf. of the max. number of retransmissions.

Figure 5: Evaluation of the performance in terms of maximum resource occupation.

References

[1] Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications. *ETSI TC ERM, TR 103 055, v1.1.1*, September 2011.

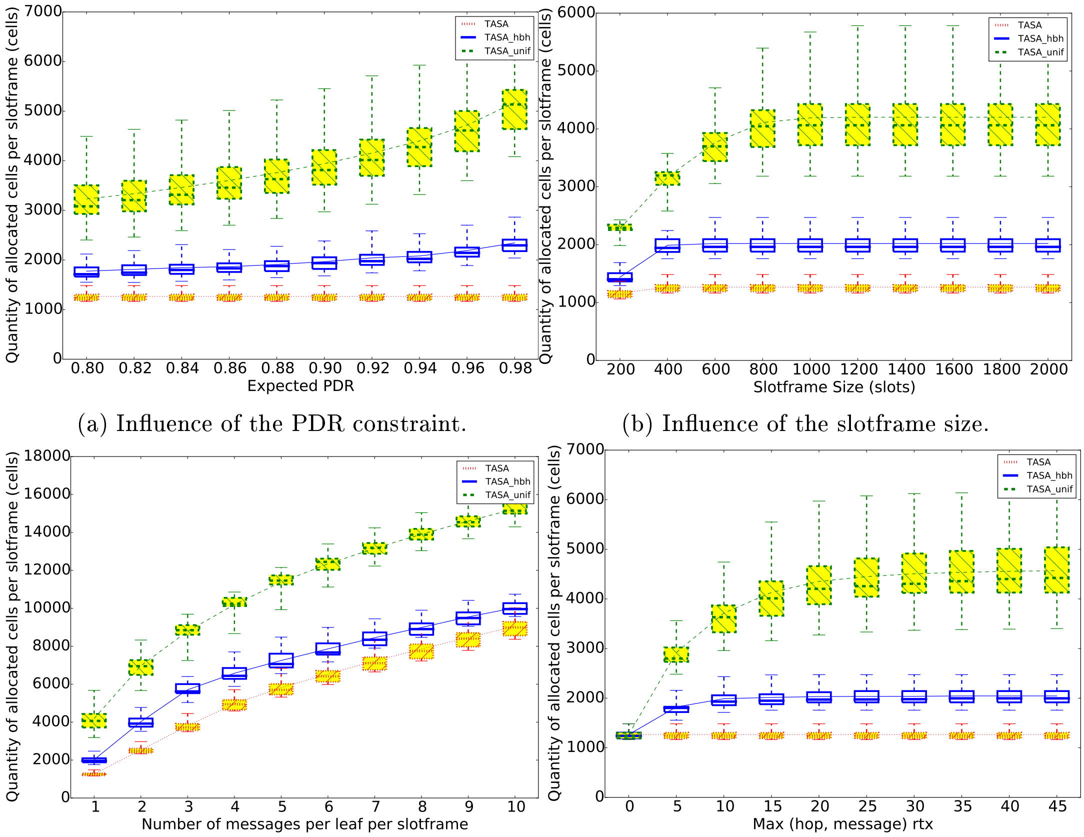
[2] Nouha Baccour, Anis Koubâa, Luca Mottola, Marco Antonio Zúñiga, Habib Youssef, Carlo Alberto Boano, and Mário Alves. Radio Link Quality Estimation in Wireless Sensor Networks: A survey. *ACM Transactions on Sensor Networks*, 8(4):1–33, September 2012.

[3] F. Dobslaw, T. Zhang, and M. Gidlund. End-to-end reliability-aware scheduling for wireless sensor networks. *Industrial Informatics, IEEE Transactions on*, PP(99):1–1, 2014.

[4] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[5] Maria Rita Palattella, Nicola Accettura, Mischa Dohler, Luigi Alfredo Grieco, and Gennaro Boggia. Traffic-aware time-critical scheduling in heavily duty-cycled ieee 802.15. 4e for an industrial iot. In *Proc. of IEEE Sensors 2012*, pages 1–4, 2012.

[6] Maria Rita Palattella, Nicola Accettura, Luigi Alfredo Grieco, Gennaro Boggia, Mischa Dohler, and Thomas Engel. On optimal scheduling in duty-cycled industrial iot applications using ieee802. 15.4 e tsch. *Sensors Journal, IEEE*, 13(10):3655–3666, 2013.



(a) Influence of the PDR constraint.

(b) Influence of the slotframe size.

(c) Influence of the traffic load.

(d) Inf. of the max. number of retransmissions.

Figure 6: Evaluation of the performance in terms of network resource usage.

[7] M.R. Palattella, N. Accettura, L.A. Grieco, G. Boggia, M. Dohler, and T. Engel. On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH. *Sensors Journal, IEEE*, 13(10):3655–3666, 2013.

[8] P. Thubert, T. Watteyne, M.R. Palattella, X. Vilajosana, and Qin Wang. Ietf 6tsch: Combining ipv6 connectivity with industrial performance. In *IMIS*, pages 541–546, July 2013.

[9] T. Winter, P. Thubert, and al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, IETF, March 2012.

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399